Growing Surface Structures

Hendrik Annuth and Christian-A. Bohn
Computer Graphics & Virtual Reality, Wedel University of Applied Sciences, Feldstr. 143, Wedel, Germany

Keywords: Unsupervised Learning, Competitive Learning, Growing Cell Structures, Surface Reconstruction, Surface Fitting.

Abstract: Strictly iterative approaches derived from unsupervised artificial neural network (ANN) methods have been surprisingly efficient for the application of surface reconstruction from scattered 3D points. This comes from the facts, that on the one hand, ANN are able to robustly cluster samples of arbitrary dimension, size, and complexity, and on the second hand, ANN algorithms can easily be adjusted to specific applications by inventing simple local learning rules without loosing the robustness and convergence behavior of the basic ANN approach.
In this work, we break up the idea of having just an "adjustment" of the basic unsupervised ANN algorithm but intrude on the central learning scheme and explicitly use learned topology within the training process. We demonstrate the performance of the novel concept in the area of surface reconstruction.

1 INTRODUCTION

Due to the rapid development in 3D scanning technology, real world objects can be scanned faster, more accurately and at a higher resolution. This allows creating high quality virtual representations of these objects that can be utilized for many different purposes in digital data processing. In archaeology and crime scene investigation a site can be analyzed independently from its location and time. Architecture and plant manufacturing involve accurate construction planning and precise measuring, which can be efficiently done on a digital model of a construction site. In films reconstructed surfaces are often used as a support structure for manually created models. Computer games use 3D scanning technology to incorporate real world objects and environments into their otherwise virtual worlds. Reverse Engineering, the most important field of reconstruction, is a process where exiting products or physical prototypes are used as a template to implement a virtual 3D model for production and quality control purposes. Mobile robots can profit from surface models as maps for path planning, localization, scene interpretation and grasping.

Laser scanning devices take samples of present surfaces as three dimensional data points. These points accumulate as an unorganized cloud of points. Surface reconstruction is the process which creates a virtual model of the surfaces from which these points originate (see Fig. 1). Such a point set typically includes noise, outliers, non-uniform sample densities, and holes. Since the combination of these problems is often inherently ambiguous, a vast number of different reconstruction approaches, and pre- as well as post-processing methods exist.

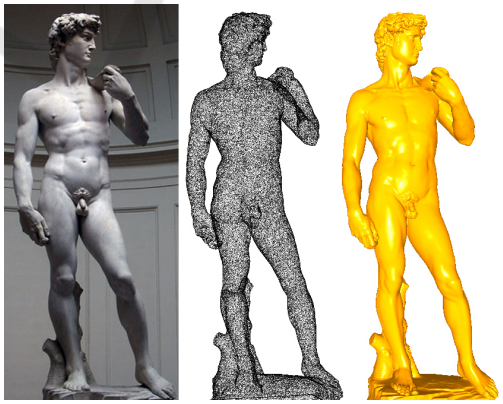

Figure 1: A photography of Michealangelo's David (left), a point cloud of the David statue (middle), and a surface fitted into that point cloud (right).

Due to the remaining challenges in the field of reconstruction there is a strong tendency to focus on ANN based solutions since they are strong with incomplete and noisy data while being flexibly adaptable. Thus, there is hope that by a more intuitive, "ad-hoc" manner, ANN training can be modified to match the problems under consideration without the need of

Annuth H. and Bohn C..
Growing Surface Structures.
DOI: 10.5220/0004529203490359
In *Proceedings of the 5th International Joint Conference on Computational Intelligence* (NCTA-2013), pages 349-359
ISBN: 978-989-8565-77-8

a deterministic mathematical model.

2 PREVIOUS WORK

2.1 Classical Surface Reconstruction

Many surface reconstruction approaches have been suggested over the last decades. Range image methods can achieve very high resolutions and accuracy, while necessitating a very controlled scanning setup with a limited sensing area (Curless and Levoy, 1996). Region growing approaches (Gopi and Krishnan, 2002; Bernardini et al., 1999) extended an initial surface incrementally at its boundaries. Some methods reduce a 3D Delaunay triangulation of the samples to a final surface (Edelsbrunner and Mücke, 1992), some derive it from the Voronoi diagram of the points (Mederos et al., 2005; Amenta et al., 1998). Combined concepts use region-growing approaches for the triangulation and an additional global graph like a 3D Delaunay triangulation as a guidance (Kuo and Yau, 2005) or a *medial scaffold* (MS) (Chang et al., 2009). Balloon models construct a volumetric object surface by the "inflation" of a small surface, as if it would be a balloon, inside a point cloud (Sharf et al., 2006). Another huge class of reconstruction methods demands points that are augmented by its normals to define an incomplete distance function. This function is completed and the subspace in R^3 for which it returns zero — the zero-level-set — is the surface. The function can be composed of a multitude of linear functions (Hoppe et al., 1992), quadratic functions (Kazhdan et al., 2006; Ohtake et al., 2003) or radial base functions (Carr et al., 2001). Model based reconstruction approaches compose a surface of a multitude of predefined models or components, which are recognized and fitted into the point cloud (Schnabel et al., 2009; Gal et al., 2007). Warping algorithms approximate the surface by deforming an initial surface to match the given points (Yu, 1999; Baader and Hirzinger, 1993).

2.2 Artificial Neural Network based Reconstruction

Many neural computation techniques have been applied to the problem of surface reconstruction and are based on unsupervised learning concepts. Algorithms such as the *k*-means clustering approach (MacQueen, 1967) use reference vectors to accomplish classification and clustering tasks on huge and challenging data sets ("hard competitive learning"). Kohonen presented the *Self-Organizing-Map* (SOM) (Kohonen, 1982) — additionally reference vectors are connected adding a topology ("soft competitive learning") which enables the construction of a surface over the sample set. Kohonen's approach has the disadvantage of a fixed resolution, which strongly relates the results to the initial setting and size of the network. Fritzke presented the *Growing-Cells-Structure* (GCS) approach (Fritzke, 1993) where the network grows over time by dynamically adding reference vectors. The growing process can be determined by the approximation error toward a likelihood distribution or a quantization error, both of which are measured in relation to the reference vectors. Based on this algorithm many convincing surface reconstruction — sometimes referred to by the term "refinement strategies" — methods have been presented (Annuth and Bohn, 2012; Ivrissimtzis et al., 2003a; Ivrissimtzis et al., 2003b; Vrady et al., 1999). The main disadvantage of the methods above is the fact that 2D subspaces or surfaces are approximated by point distributions instead of surface models. This becomes most apparent when modeling flat surface areas where the granularity of the ANN surface depends on the distribution of samples and not on the complexity of the underlying surface.

In this work we present an approach which lifts this handicap. Basic ANN learning is changed to a surface oriented learning saving the advantages of neural networks but concurrently implementing a reasonable "surface learning". The difference to former approaches which modify ANNs by adding additional constraints to the learning rules our approach introduces an actual novel learning scheme.

3 SURFACE RECONSTRUCTION

Surface reconstruction creates a 2D subspace $\mathcal{S}$ in a 3D space $\mathbb{R}^3$ that represents a real world physical surface $\mathcal{S}_{phy}$. The information given about $\mathcal{S}_{phy}$ is a finite collection of surface samples $\mathcal{P} = \{p_1 ... p_n\} \subset \mathbb{R}^3$. If closest neighbors in $\mathcal{P}$ always indicate a connection on $\mathcal{S}_{phy}$, surface neighborhood relations can be investigated by accessing $\mathcal{P}$ in a 3D search pattern. Real world scenarios however, involving noise, non-uniform sample densities and incompletely sampled areas. This makes a 3D search unreliable (see Fig. 2), which is the basic problem to overcome in a reconstruction approach. Note that many of the following illustrations are in 2D and are therefore curve reconstructions, but to avoid confusions by swapping terminology, we proceed in using the terms of 3D surface reconstruction.

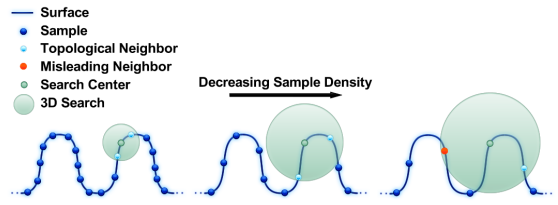

Figure 2: The topology of a surface has to be derived from 3D samples. But searching in 3D for sample neighbors might produce misleading results. First, a highly sampled surface, where the two surface neighbors to a certain sample can be easily found (left). Then the same surface with a lower sampling, the search space for the two closest neighbors has grown (middle). And at last, a low sampling where the two closest neighbors are not the correct topological (on surface) neighbors (right).

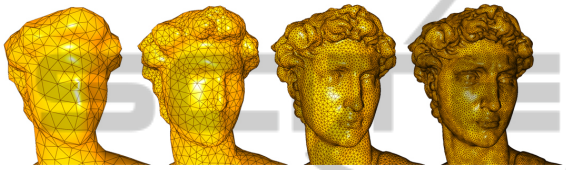

Figure 3: Different successive surface stages in a GCS surface reconstruction.

4 GROWING CELL STRUCTURES

A GCS network is composed of simplices of an initially chosen dimension. In case of surface reconstruction, a 2D surface $\mathcal{S}$ built of triangles as simplices. The initial surface is a very simple network of triangles such as a tetrahedron. It is positioned roughly at the center of $\mathcal{P}$. Since the GCS algorithm is inspired by growing organic tissue, the reference vectors are termed cells. GCS use an iterative refinement process to fit the current surface into the point data $\mathcal{P}$ (see Fig. 3). The refinement process randomly selects a sample of $\mathcal{P}$ and deforms the current surface in order to progressively minimize its distance. This basic step is repeated and in each iteration the local approximation error is measured. These errors are used to determine surface areas which need to be refined by local subdivision processes. Subdivision and further iterations lead to a better match of the surface to the sample distribution and the process is stopped when the chosen average approximation error reaches a threshold or a certain number of reference vectors is reached. In the following section we analyze the algorithm in detail for the application of surface reconstruction and show different kinds of handling the error such as likelihood distribution, error minimization and topology preservation which each lead to different results. Since the algorithm represents $\mathcal{S}$ as a triangular mesh, we will use the term vertex instead of reference vector, since it is more common in this context.

Algorithm 1: An overview of the GCS algorithm. Conditions 1 and 2 can be defined as simple counters.

1: Given a point cloud $\mathcal{P} = \{p_1 ... p_n\}$ and an initial surface $\mathcal{S}$ in form of a tetrahedron represented as an interconnected network of vertices $\mathcal{S} = \{v_1 ... v_n\}$.
2: **repeat**
3: **repeat**
4: **repeat**
5: Select random sample p_x of $\mathcal{P}$ and search the winning vertex v_x with smallest Euclidian distance to p_x.
6: Move v_x towards p_x.
7: Move all direct neighbors of v_x with a lesser factor towards p_x.
8: Adapt the approximation error of v_x.
9: **until** condition 1 holds.
10: Add new vertex in the area of highest approximation error through a vertex split operation.
11: **until** condition 2 holds.
12: Search the least winning vertex in the network and delete it by an edge collapse operation.
13: **until** accuracy exceeds a certain threshold.

4.1 Likelihood Distribution

The approximation error (algorithm 1 line 8) can be altered toward a likelihood distribution or a quantization error (see section 4.2). The vertices create a likelihood distribution if for every given vertex $v \in \mathcal{S}$ the likelihood to be the closest neighbor to a randomly chosen sample $p \in \mathcal{P}$ is equal. If we see vertices being closest neighbors as the result of a probability experiment this approximation resembles entropy maximization. This means that the information carried by any given sample is of the same importance. These representations are especially important in pattern recognition and statistical analysis.
To achieve this every vertex carries a signal counter. To approximate the likelihood distribution (algorithm line 8) these counters are simply incremented when a vertex is closest to an input sample. If a new vertex is added (algorithm line 10) the highest error term refers to the space where most samples share the same vertex.
Since older signals tend to be less representative, all signal counters are decreased at every iteration cycle by a certain factor (algorithm line 8). By using a likelihood distribution signal counters can also be used

to determine misplaced vertices in spaces that contain few or even no samples, since their signal counters are very low due to constant decreasing. This concept has therefore been used in most implementations for surface reconstruction.

These algorithms however determine the area for which the likelihood of the vertices is highest and not the surface. If a flat surface is approximated, the algorithm will create lots of vertices in relation to the amount of samples, although the area could be accurately approximated with a few triangles only.

4.2 Distance Minimization

When the approximation error (algorithm line 8) is changed to account for a quantization error, vertices are placed exposing the smallest Euclidian distance to the samples in $\mathcal{P}$. If the samples $\mathcal{P}$ are equally distributed the goals of a likelihood distribution compared to an error minimization are nearly the same. If however some regions are represented by a denser sampling than others, these regions will be represented by less vertices in the error minimization scenario, since the error which is measured as the Euclidian distance can be lowered more significantly in regions where samples lay farther apart, hence vertices are more likely to be added there. This approximation is typically used for vector quantization in data compression. To implement this behavior every vertex carries an error value which is increased (algorithm line 8) by the distance or the squared distance between the winning vertex and the given sample. The highest approximation error refers to the space where the samples lay farthest away from a vertex, thus a new vertex will be added there (algorithm line 10). In contrast to a likelihood value, removing a vertex with low distance errors would make no sense, since these vertices indicate that they are well placed. However in case the created topology matters, as in surface reconstruction, it is reasonable to remove such vertices for memory efficiency reasons, since they might be redundant geometry wise.

The basic problem is the difference between the approximation of the right topology and achieving a lowest distance error. We will discuss this problem in more detail in section section 4.3. When this approximation error is used, the deletion process of misplaced vertices need to be handled separately. Despite of this disadvantage minimizing the distance error might be more convenient for surface reconstruction. But this is not the case if the approximation minimizes the distance to the vertices instead of the surface (see Fig. 4), since a surface approximation aims to fit $\mathcal{S}$ as close a possible to $\mathcal{P}$.

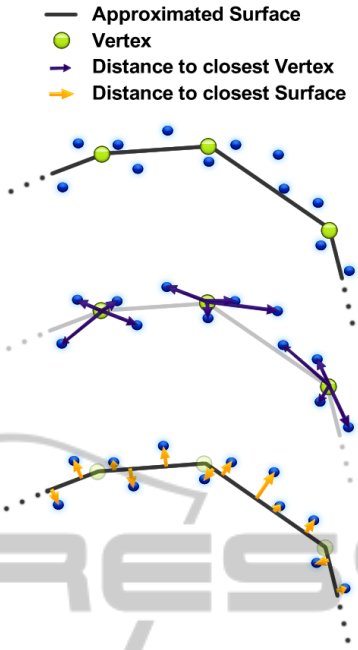

Figure 4: Samples and an approximated surface (top), distance between vertices and samples (middle), distance between surface and samples (bottom). In case of surface approximation the distance to the surface is obviously more worthy.

Many implementations have tackled this problem indirectly. (Annuth and Bohn, 2012) presents a roughness adaptation where the average surface curvature is compared to the one of a winning vertex and curved areas lead to higher signals leading to more subdivisions in such areas. In (Jeong et al., 2003) vertices additionally have normals and the algorithm counts how much these normals are moved to increase subdivisions in such areas.

These changes lead to an implicit representation of the approximation error within the algorithm, since curved surface regions need more subdivisions to be correctly approximated. But the surface approximation error itself is not explicitly represented.

4.3 Topology Preservation

The SOM (Kohonen, 1982) introduced an unsupervised learning concept with an additional topology. In the given network $\mathcal{S}$ vertices are not allowed to move independently. When a vertex is moved toward a sample its neighbors are also moved to decrease the created surface tension (algorithm line 7). This principal adds elasticity to the network — the behavior of a continuous surface is manifested implicitly by creating dependencies between the vertices. A topology can increase the performance in placing ref-

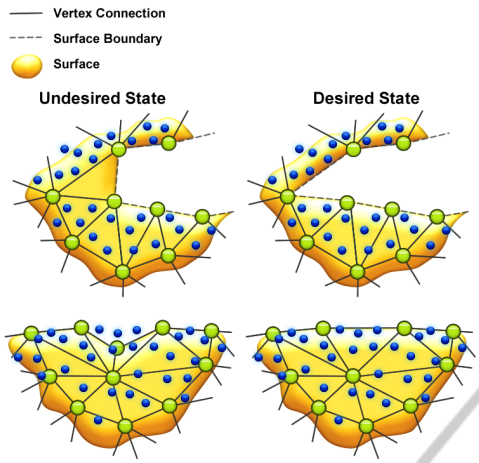

Figure 5: Two surfaces with the same vertices, the same samples, and with the same approximation error, that expose an undesired (left) and a desired (right) solution. A triangle placed in an empty space (top) and an incorrect dent in the surface (bottom).

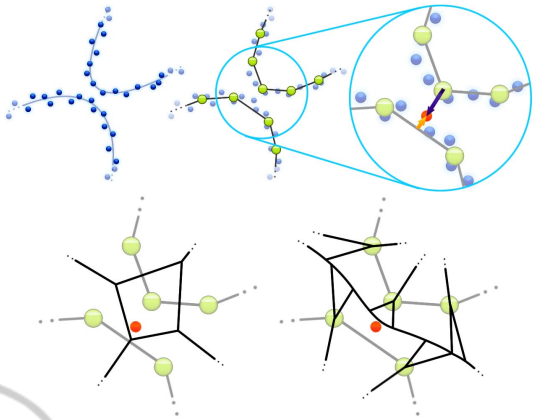

Figure 6: Samples that originate from a curved surface (top,left), a fitting approximation of this surface and a magnification that shows the vertex-sample distance and the surface-sample distance (top,right), the vertex Voronoi regions which assign a sample at the wrong surface (bottom,left), the surface Voronoi regions which assign the same sample correctly (bottom,right).

erence vectors since the dependencies between them make their movements more stable and thereby make smoother distributions more likely. But the created topology itself can be used in many different ways as well. Data of a high dimensional input space can be mapped into a space of lower dimension and can then be visualized or analyzed with less computational effort (dimensionality reduction). The topology can also be used for regression analysis where $\mathcal{P}$ is known to originate from an unknown continuous function to be reconstructed from the data (function approximation). The SOM uses a static topology, which usually resembles a square shaped grid. The standard GCS algorithm also uses a static surface topology while the connectivity of the network can change (note that the network connectivity is often also referred to as topology, perceiving the network as a graph). This means a network area can be increased in resolution and thereby gather new vertices and connections, but the surface topology of a sphere, inherited from the initial tetrahedron shape, cannot be changed.

So functions that have a different topology can not be correctly resembled. GCS however has better surface approximation capabilities than the SOM, since it builds newly created surfaces by refining a former version of that surface. This adapts the vertex resolutions in different surface areas toward the target function and gives a surface a certain inertia when being modified, which avoids local failures if a surface is fitted into a challenging point constellation.

If the information of a sample placed in a 3D space is processed by the algorithm, this information is always set considering a pre-existing current surface $\mathcal{S}$. This is the strategy of the GCS algorithm to overcome the 3D search problem (see section 3). But still the GCS can get stuck in local minima and the initial topology can mismatch the target surface.

In the standard algorithm a destructive method is presented which uses the average edge length as an indicator to cut out triangles (Fritzke, 1993). In (Ivrissimtzis et al., 2003a) triangles which are larger than the average size are cut out, boundaries that fall below a certain Hausdorff distance are joint, and in (Annuth and Bohn, 2012) high valences are used as an indicator to cut the surface and low distances in comparison to edge length to join boundaries. With these changes complex topologies can be created. The main problem of all presented GCS based algorithms concerning topology issues is the missing representation of the actual surface within the adaptation process, which makes many insufficient approximation states simply not measurable (see Fig. 5).

Even so the given 3D information of samples is set in relation to the existing 2D surface, the surface is still represented as a collection of Voronoi regions of the vertices, since vertex-sample and not surface-sample distances are considered. This concept implicitly includes the assumption that the Voronoi regions of two connected vertices will not be interrupted within their attached surface. But for close or complex shaped surfaces, this is not the case (see Fig. 6). Here the actual representation of $\mathcal{S}$ becomes apparent being a permeable space of independent Voronoi volumes.

Topology preservation and error minimization are two different things. Error minimization tries to create the smallest possible distance to the samples in $\mathcal{P}$. Topology preservation however is concerned with reaching

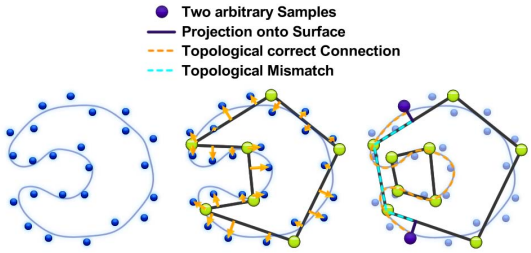

Figure 7: A surface and samples (left), an approximation of that surface and the distance error between the surface and the samples (middle), another approximation of that surface with a topological error (right).

a topology with $\mathcal{S}$ that lies as close as possible to the topology of $\mathcal{S}_{phy}$ (see Fig. 7).
In order to be topologically correct every point on one surface need to have a unique equivalent on the other surface and vice versa, while neighbor relations are preserved, meaning the shortest on-surface path between any two given points projected onto $\mathcal{S}$ should always correspond to the one on $\mathcal{S}_{phy}$.

5 APPROACH

The GCS algorithm has proven to be a high quality surface reconstruction tool. However, in our analysis of the algorithm we saw that topology is only created implicitly and only accounted for through the additional adjustment of neighboring vertices. In the following section we will present our changes to the general approximation concept of the basic algorithm and then the improvements that can be made based on these changes.

5.1 Topology focused Approximation

The basic algorithm concept focuses on placing vertices in positions likely to decrease the chosen approximation error. To put the actually created surface topology into focus, the approximation error needs to be set in relation to the surface-sample distance. The most important change is to search for the closest surface element (algorithm line 5) instead of the closest vertex. The adaptation process (algorithm line 6 and 7) can now also be set in relation to a sample being closest to a triangle or being closest to an edge, which gives rise to more different local surface modifications (see section 5.2.2). In the basic GCS implementation the signal counter or error value is carried by the vertices. The surface structure element that most distance errors are measured towards and that is also the building block of the discretization of $\mathcal{S}_{phy}$ is the triangle and is therefore the structure that carries the local approximation error values in our implementation. The most sensible place for the error value of any topology focused function approximation is always the simplex of highest dimension in the GCS algorithm. The distance between a sample and its closest structure represents the actual distance error to the approximated surface and gives this approximation error way more validity (algorithm line 8). This allows for better judgments about a current local approximation state and the choice of location for subdivision (algorithm line 10). The new approximation error also allows and demands to distinguish between topology changing deletions to correct topologically misplaced surface structures and non-topology changing deletions that remove geometrically redundant structures from the surface (algorithm line 12). Topology changing deletions are realized by adding an "age" to every triangle and cutting them out when they reach a certain age.

5.2 Adaptations of the Algorithm Behavior

With the changes described in section 5.1 additional and more accurate information about the current approximation state is available within the GCS process. This information can be used to create a better approximation result in accuracy and topology. In the following we will present our implementation details.

5.2.1 Search for Closest Element

Due to run time efficiency reasons we did not use an actual triangle based spatial subdivision data structure, but still a vertex based octree. By searching for a number num_v of vertices and checking their surrounding triangles, we heuristically find the closest structure to a given sample. We used $num_v = 3$ which fails when the degree of curved and flat surface areas diverge too much and are close to each other, but we consider this case to be rare (see Fig. 8). The new search process has three possible outcomes: a vertex, an edge, or a triangle (algorithm line 5).

5.2.2 Surface Movement

Instead of having only a vertex as a closest element, we now can access additionally an edge and a triangle in our implementation. We modeled three different main movements (algorithm line 6). Since we now know the distance of a given sample p to the surface we can compare this distance d_p to the average sample-to-surface distance $\overline{d_p}$. When d_p is only

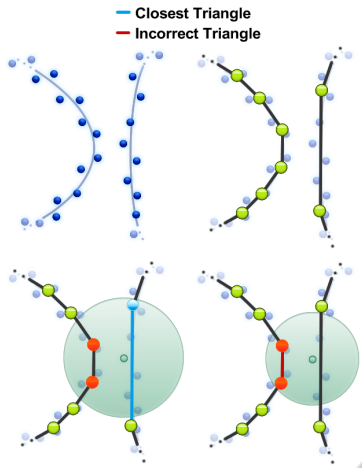

Figure 8: The presented approach searches for the closest surface structure, such as a vertex, an edge or a triangle. However this process is emulated on a search investigating the surrounding elements of a number of vertices num_v, instead of having a search tree actually comprising edges and triangles. The figure shows a sampled curved and flat surface close by (top, left), next to it its approximated surface (top,right). The search heuristic works correctly for $num_v = 3$, where three vertices are investigated (bottom,left). The search heuristic fails for $num_v = 2$ (bottom,right), where the vertex connected to the closest triangle is not investigated.

a fraction lim_{skip} of $\overline{d_p}$ we entirely discard the adaptation since the sample already lies more or less close to the surface. When d_p is about the degree lim_{single} lower than $\overline{d_p}$ we only move the closest vertex, since we consider the surface to be generally correct, but the "joints" of the divisions can be optimized. We use $lim_{skip} = 0.9$ and $lim_{single} = 1.2$. Note that distance error distribution within the process is not Gaussian, thus we cannot describe the process in terms of standard deviations. For higher distances we move all vertices of the given structure towards the sample. Due to this, surface widening is less likely to cause spikes and the surface is moved more unified, which produces a smoother surface. The neighborhood movement (algorithm line 7) is unchanged and accomplished with the Laplace smoothing mechanism (Taubin, 1995) for all first neighbors of the given structure.

5.2.3 Distance Error

A sample can be closest to a vertex, an edge, or a triangle. When it is closest to a triangle the age of this triangle is set to zero and the distance error is changed. In case of an edge this is done for both triangles connected to that edge. In case of a vertex it is done for all triangles connected to this vertex. If

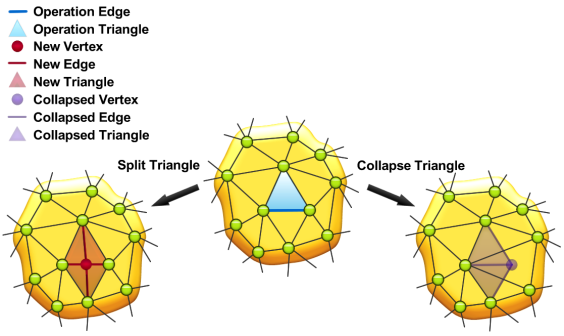

Figure 9: A triangle split operation (left) and a triangle collapse operation (right).

we would set the error value directly to the given distance, all previous distance errors would be lost, if we, on the other hand accumulate all distance errors, old distance errors would totally determine the refinement process, since their distances where huge and it would take many subdivisions to decrease them. We could constantly decrease all error values as described in section 4.1 but this would strongly decrease the accuracy of the local distance errors since it implies that the surface constantly improves everywhere with a constant rate. We wanted the half-life λ of a distance error to be 9 winning operation (see equation 1).

$$\begin{aligned} \left(\frac{k-1}{k}\right)^{\lambda} &= 0.5 \\ k &= \frac{-1}{\sqrt[\lambda]{0.5}-1} \end{aligned} \qquad (1)$$

$$err_{new} = \frac{d_p + err_{old}(k-1)}{k}$$

5.2.4 Refinement

Instead of the vertex we search the *triangle* with the highest approximation error. Subdivision is done by splitting the triangle's surface from one of its three vertices to the opposite edge and then also split the other triangle in the mesh with this edge. The edge with the additional triangle with the largest error term is taken. Four new triangles are added, four new edges, and one new vertex (see Fig. 9). The error value of a new triangle is the half of the error value of its predecessor.

5.2.5 Deletion

The deletion process is one of the most important changes in the algorithm. When using a sample-to-surface distance error, the error values can only be used to determine triangles that might be redundant and do not improve the approximation. In order to have a model representation that is as memory efficient as possible, these triangles can be deleted by an edge collapse operation of one of its three edges. The

best candidate for the operation is the edge which is surrounded by triangles with those normals that expose the least differences $|\nabla n|$ to one another, since a collapse of this edge will change the surface gradient the least. This edge can be determined as the one with the highest dot product of the normals of its two vertices, since these normals are calculated based on their surrounding triangles. It is reasonable to set a threshold $max_{|\nabla n|}$ for $|\nabla n|$ to avoid changes that decrease the surface approximation quality. We choose $max_{|\nabla n|}$ to be $cos\ 10^\circ$ which allows for a maximum angles of 10° between those normals. In addition to the surface error we need a triangle age a that indicates if a triangle reached a maximum age max_a. This should happen when the triangle has been missed for a certain number of times η. Those triangles are considered to be misplaced and topological wrong. A misplaced triangle is detached from the rest of the network and then deleted. Note that the deletion of a single triangle can leave the surface in an undesired state, where for example triangles are connected by single vertices only, which need to be cleared. Since triangles are likely to have different sizes and small triangles are less likely to be winners, the real age a_{real} of a triangle need to be set in relation to its size, or in other words, small triangles age slower. This can easily be achieved by an additional age factor t_{size} calculated as the size of a triangle divided by the average triangle size. For small triangles this value is very low, so their age is strongly reduced. For every iteration the age of all triangles is increased by a tiny factor μ that has a relation to the overall number of triangles $|\mathcal{T}|$ (see equation 2). This is basically the reversed tumble tree (Annuth and Bohn, 2010) principal. We use 1 for max_a and 10 for η. This means, *i.e.*, if one of four triangles of the same size is not hit one time for forty iterations it is considered to be misplaced. The deletion process now explicitly distinguishes distance driven and topologically driven deletion.

$$\begin{aligned} \mu^{(|\mathcal{T}|\cdot\eta)} &= max_a \\ \mu &= \sqrt[(|\mathcal{T}|\cdot\eta)]{max_a} \\ a_{new} &= a_{old} \cdot \mu \\ a_{real} &= t_{size} \cdot a \end{aligned} \tag{2}$$

5.2.6 Finalization

One of the assets of the GCS algorithm is the fact that $\mathcal{S}$ is an approximation of $\mathcal{S}_{phy}$ at any time during the running loop — the algorithm can be stopped and resumed at any given time. With the novel surface distance approximation error a potential stopping point for the algorithm can be chosen more sensibly.

Figure 10: A progression series of the dragon model from left to right with 2500, 5000 and 10000 triangles with the standard algorithm (top) and with the new algorithm (bottom). With the new algorithm the surface diverges faster toward the final topology.

6 RESULTS

We accomplished different tests with the Stanford Dragon model as a good example for a point cloud that is relatively challenging by its shape and the sample distribution, the hand model exposes sharp features, the Asian Dragon and the Thai Statue expose a lot of curved areas, the Heating Pipes model includes some extremely noisy areas, non-uniform sample densities and open surface areas, the Happy Buddha has regions of surfaces lying close together. From the basic algorithm we used (Annuth and Bohn, 2012) but deactivated roughness adaptation and sharp feature detection.

Theoretically, any surface would be correctly reconstructed with the GCS algorithm, if infinite samples, sufficient memory and time were available. A reasonable parameter to judge efficiency is the time to reach a certain accuracy. In our experiments with the new approach, for instance the topology of the dragon model was approximated much faster, shown in Fig. 10.

Close surfaces can be handled correctly with our presented method and a vast number of additional iterations to avoid permeating Voronoi regions is not required any longer as we show in Fig. 11.

Although the standard algorithm is already quite robust when dealing with noise, we could show that spikes and rough surface gradients could be greatly reduced with our presented method. The moving of entire substructures seems to have a smoothing effect on the surface (see Fig. 12).

We compared the new and the old algorithm (see table 1). Generally the old algorithm creates a lower average point $\mathcal{P}$ to surface $\mathcal{S}$ distances, since it evenly distributes its subdivisions over $\mathcal{S}$, whereas the presented algorithm focuses its subdivisions on areas of high ap-

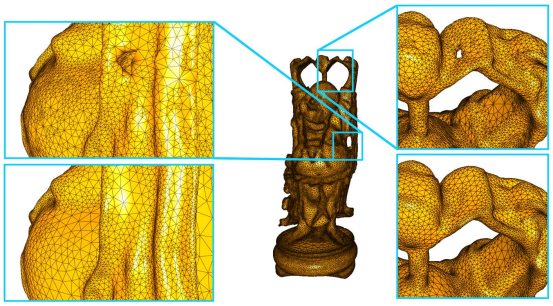

Figure 11: Some thin areas of the Happy Buddha model, reconstructed with 200K triangles with the old (top) and the new (bottom) algorithm. The new algorithm is able to build a correct topology in thin areas in an earlier algorithm stage.

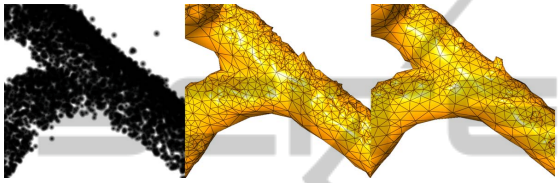

Figure 12: Very noise section of the Heating Pipes model (left); Pointy vertices or spikes on the surface of the standard algorithm (middle) and a smoother surface with our approach (right).

proximation error rates. This is visible through a 25% decrease of the mean squared error. Especially for curved models such as the Asian Dragon and the Thai Statue this effect is very salient.

Although the search process is more complex, the extra time costs are nearly leveled by the discarded operations which are for the Dragon model 43.3% of discarded adaptations and a rate of 26.3% inside the surface movements of vertices only with our setting for lim_{skip} and lim_{single}.

When we used the square distance as the approximation error, the results for both the average distance error as well as the square distance error where worse, than the results of the standard algorithm. This is reasonable since most triangles are used up to model tiny but steep curvature (see Fig. 13). In addition, triangles tended to clump even for low error half-lifes λ. For the Happy Buddha model this led too many clump-like artefacts. We consider this setting generally impractical.

Test Hardware A *Dell*®Precision M6400 with *Intel*®QX9300 (2.53GHz) processor with 8GB 1066 MHz DDR3 Dual Channel RAM.

Parameter Settings $num_v = 3$; $max_a = 1$;
$\eta = 10$; $max_{|\nabla n|} = cos\ 10°$; $\lambda = 9$;
$lim_{skip} = 0.9$; $lim_{single} = 1.2$;
(See (Annuth and Bohn, 2012) for the parameter settings of the basic algorithm).

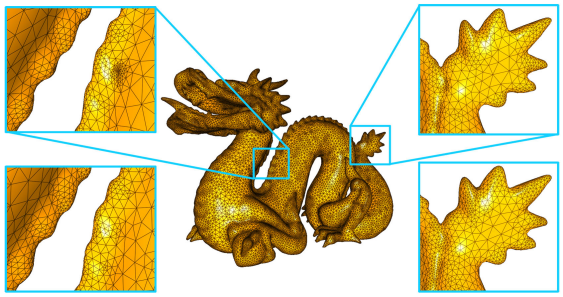

Figure 13: Two magnifications of the dragon model reconstructed with100K triangles with the square of the point to surface distance as the approximation error d_p^2 (top) and with just the distance error d_p (bottom). In the latter case, the triangle resolutions at the back and between tail and body are very high and are even for curved areas overrepresented.

7 CONCLUSIONS AND FUTURE WORK

In this paper we focused on the behavior of the growing cell structures approach as a function approximation algorithm. We analyzed the GCS with the classical adaption algorithm for matching requirements of surface reconstruction. Derived from these observations we presented our new GCS learning model and proved theoretically and by examples that it outperforms the classical GCS approach. The basic idea of the presented approach is to incorporate the constructed topology into the GCS learning scheme. GCS creates ideal distribution matching, clustering, or dimensionality reduction by an *implicit* representation of a topology. In our work, we introduced an *explicit* topology to model the approximation behavior according to it, while saving the valuable ANN capabilities mentioned above. As result, we got a new ANN learning strategy, which showed several advantages compared to classical models. We see this paper as proof of our conceptual change of the GCS algorithm, giving rise to many improvements for the algorithm in future work.

ACKNOWLEDGEMENTS

The authors would like to thank the Stanford University Computer Graphics Laboratory and the Hamburg Hafen City University for making their laser scanned data available to us. We also would like to thank Kai Burjack for his assistance in rendering.

Table 1: Our results for different models. We expose the time for the reconstruction process (time), the average distance to $\mathcal{P}$ (dist) times 10^4 and the square distance to $\mathcal{P}$ (dist^2) times 10^7. We also tested different values for num_v. Note that models are normalized by setting the cubic diagonal of their bounding box to one.

	GCS			GCS error=d_p			GCS error=d_p^2		
Model (# triangles)	time[s]	dist	dist^2	time[s]	dist	dist^2	time	dist	dist^2
Hand (20K)	6	3.19	4.38	7	4.29	4.11	6	4.70	5.00
Dragon (100K)	61	2.29	3.42	65	2.55	2.97	66	3.05	4.34
Asian Dragon (100K)	55	2.36	3.62	61	2.71	2.53	61	2.83	2.78
Thai Statue (200K)	146	3.21	15.4	147	3.00	4.02	148	4.05	7.17
Happy Buddha (200K)	150	1.48	14.1	158	1.89	11.7	160	4.02	63.0
	$num_v = 1$			$num_v = 5$			$num_v = 10$		
Model (# triangles)	time[s]	dist	dist^2	time[s]	dist	dist^2	time	dist	dist^2
Dragon (100K)	62	2.35	3.01	71	2.55	3.05	121	2.57	2.95

REFERENCES

Amenta, N., Bern, M., and Kamvysselis, M. (1998). A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 415–421, New York, NY, USA. ACM.

Annuth, H. and Bohn, C. A. (2010). Tumble tree: reducing complexity of the growing cells approach. In *Proceedings of the 20th international conference on Artificial neural networks: Part III*, ICANN'10, pages 228–236, Berlin, Heidelberg. Springer-Verlag.

Annuth, H. and Bohn, C.-A. (2012). Smart growing cells: Supervising unsupervised learning. In Madani, K., Dourado Correia, A., Rosa, A., and Filipe, J., editors, *Computational Intelligence*, volume 399 of *Studies in Computational Intelligence*, pages 405–420. Springer Berlin / Heidelberg. 10.1007/978-3-642-27534-0_27.

Baader, A. and Hirzinger, G. (1993). Three-dimensional surface reconstruction based on a self-organizing feature map. In *In Proc. 6th Int. Conf. Advan. Robotics*, pages 273–278.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G., and Member, S. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5:349–359.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 67–76, New York, NY, USA. ACM.

Chang, M.-C., Leymarie, F. F., and Kimia, B. B. (2009). Surface reconstruction from point clouds by transforming the medial scaffold. *Comput. Vis. Image Underst.*, 113(11):1130–1146.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA. ACM.

Edelsbrunner, H. and Mücke, E. P. (1992). Three-dimensional alpha shapes. In *Volume Visualization*, pages 75–82.

Fritzke, B. (1993). Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7:1441–1460.

Gal, R., Shamir, A., Hassner, T., Pauly, M., and Cohen-Or, D. (2007). Surface reconstruction using local shape priors. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, SGP '07, pages 253–262, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Gopi, M. and Krishnan, S. (2002). A fast and efficient projection-based approach for surface reconstruction. In *Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, SIBGRAPI '02, pages 179–186, Washington, DC, USA. IEEE Computer Society.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. A., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. In Thomas, J. J., editor, *SIGGRAPH*, pages 71–78. ACM.

Ivrissimtzis, I., Jeong, W.-K., and Seidel, H.-P. (2003a). Neural meshes: Statistical learning methods in surface reconstruction. Technical Report MPI-I-2003-4-007, Max-Planck-Institut f"ur Informatik, Saarbrücken.

Ivrissimtzis, I. P., Jeong, W.-K., and Seidel, H.-P. (2003b). Using growing cell structures for surface reconstruction. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 78, Washington, DC, USA. IEEE Computer Society.

Jeong, W.-K., Ivrissimtzis, I., and Seidel, H.-P. (2003). Neural meshes: Statistical learning based on normals. *Computer Graphics and Applications, Pacific Conference on*, 0:404.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43:59–69.

Kuo, C.-C. and Yau, H.-T. (2005). A delaunay-based region-growing approach to surface reconstruction from unorganized points. *Comput. Aided Des.*, 37(8):825–835.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. pages 281 – 297.

Mederos, B., Amenta, N., Velho, L., and de Figueiredo, L. H. (2005). Surface reconstruction from noisy point clouds. In *Proceedings of the third Eurographics symposium on Geometry processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H.-P. (2003). Multi-level partition of unity implicits. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 463–470, New York, NY, USA. ACM.

Schnabel, R., Degener, P., and Klein, R. (2009). Completion and reconstruction with primitive shapes. *Computer Graphics Forum (Proc. of Eurographics)*, 28(2):503–512.

Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L., and Cohen-Or, D. (2006). Competing fronts for coarse-to-fine surface reconstruction. In *Eurographics 2006 (Computer Graphics Forum)*, volume 25, pages 389–398, Vienna. Eurographics.

Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, New York, NY, USA. ACM.

Vrady, L., Hoffmann, M., and Kovcs, E. (1999). Improved free-form modelling of scattered data by dynamic neural networks. *Journal for Geometry and Graphics*, 3:177–183.

Yu, Y. (1999). Surface reconstruction from unorganized points using self-organizing neural networks. In *In IEEE Visualization 99, Conference Proceedings*, pages 61–64.